

Our Ref.: 51005.P110

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

**Emulation Components and System
Including Distributed Routing and Configuration of Emulation
Resources**

Inventor(s): **Frederic Reblewski**

Prepared by:

COLUMBIA IP LAW GROUP, PC
10260 SW Greenburg Road
Suite 820
Portland, OR 97223
(503) 595-2800

"Express Mail" label number EL910783908US

**Emulation Components and System Including Distributed Routing and
Configuration of Emulation Resources**

BACKGROUND OF THE INVENTION

5

1. **Field of the Invention**

The present invention relates to the field of emulation. More specifically, the present invention relates to emulation components, such as emulation integrated circuits (IC) and logic boards, emulation systems formed with such components, as well as emulation resource allocation/configuration and debugging/testing methodologies practiced therein.

10

2. **Background Information**

The first generation of prior art emulation systems were typically formed using general purpose FPGAs without integrated debugging facilities. To emulate a circuit design on one of such emulation systems, the circuit design would be "realized" by compiling a formal description of the circuit design, partitioning the circuit design into subsets, mapping the various subsets to the logic elements (LEs) of the FPGAs of various logic boards of the emulation system, and then configuring and interconnecting the LEs. The partitioning and mapping operations would be typically performed on workstations that are part of or complementary to the emulation systems, while the configuration information would be correspondingly downloaded onto the logic boards hosting the FPGAs, and then onto the FPGAs.

15

20

25

During emulation, test stimuli are either generated on the workstation or on a service board of the emulation system under the control of the workstation, and then transfer to the various logic boards for input into the emulation ICs for application to

the various netlists of the IC design being emulated. State data of various circuit elements as well as signal states of interest of the IC design being emulated, would be correspondingly read out of the applicable FPGAs, and then transfer off the logic boards, for analysis on the workstation.

5 With advances in integrated circuit and emulation technology, some late model emulation systems would employ "FPGAs" specifically designed for emulation purpose. These special "FPGAs" or emulation ICs typically would include substantial amount of on-chip reconfigurable logic elements, inteconnects, memory as well as debugging resources. As the advances continue, more and more of
10 these resources are packed into each emulation IC. As a result, more and more control signals have to be transferred onto each logic board (for transfer into the emulation ICs) to configure the emulation ICs of the logic board. Likewise, more and more state elements and/or signals of interest of the emulation ICs have to be transferred out of the emulation ICs and logic boards to facilitate analysis, leading to
15 the formation of various bottlenecks, in particular, at the logic boards, preventing efficient operation of the emulation systems.

Thus, an improved approach to forming and operating emulation systems is desired.

SUMMARY OF THE INVENTION

Data processing resources are distributively provided to an emulation system to locally and correspondingly generate configuration signals to configure selected ones of reconfigurable logic and interconnect resources of corresponding collections of reconfigurable logic and interconnect resources, to emulate corresponding partitions of an IC design. In one embodiment, the distributed data processing resources further locally and correspondingly determine interconnect routing within the selected ones of reconfigurable logic resources of the corresponding collections of reconfigurable logic resources.

In one embodiment, the distributed data processing resources are disposed on logic boards having emulation ICs that include the reconfigurable logic and interconnect resources. In other embodiments, at least some of the distributed data processing resources are disposed on the emulation ICs. The board and IC disposed distributed data processing resources cooperatively perform the earlier mentioned distributed and corresponding generation of configuration signals and interconnect routing determination.

Additionally, the present invention also contemplates the constitution of an emulation system using the aforementioned emulation ICs and logic boards.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references
5 denote similar elements, and in which:

Figure 1 illustrates the major functional blocks of a logic board incorporated with the teachings of the present invention, in accordance with one embodiment;

Figure 2a-2b illustrate a hosted emulation IC of **Fig. 1** in further details, including the on-chip debugging resources of the emulation IC in further details, in
10 accordance with one embodiment;

Figure 3 illustrates the on-board data processing resources of **Fig. 1** in further details, in accordance with one embodiment;

Figure 4 illustrates an overview of the software modules provided to the data processing resources of **Fig. 3** in further details, in accordance with one
15 embodiment;

Figure 5 illustrates an example packet suitable for use to communicate with the data processing resources of **Fig. 1**, in accordance with one embodiment;

Figure 6 illustrates an emulation system of the present invention, formed with the emulation logic board of the present invention, in accordance with one
20 embodiment;

Figure 7 illustrates the operation flow of a method of the present invention for distributively and correspondingly routing the interconnects for circuit elements of the netlists of a partition of an IC design, and configuring the reconfigurable resources of the emulation system, in accordance with one embodiment;

Figure 8 illustrates the operation flow of a method of the present invention for distributively and correspondingly pre-processing trace data of interest of an IC design being emulated, in accordance with one embodiment;

Figure 9 illustrates the operation flow of a method of the present invention for distributively and correspondingly generating and applying testing stimuli, as well as monitoring for occurrences of selected events, in accordance with one embodiment; and

Figure 10 illustrates a hosted emulation IC of **Fig. 1** in further details, in accordance with an alternate embodiment.

10

DETAILED DESCRIPTION OF THE INVENTION

In the following description, for purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Referring now to **Figure 1**, where an overview of a logic board incorporated with the teachings of the present invention, in accordance with one embodiment, is illustrated. As shown, for the illustrated embodiment, logic board **100** of the present invention includes on-board data processing resources **102**, on-board emulation ICs **104**, on-board reconfigurable interconnects **106**, on-board bus **108**, and on-board trace memory **110** coupled to each other as shown (i.e. through on-board bus **108**). Additionally, on-board emulation ICs **104** are also directly coupled to on-board trace memory **110**. Logic board **100** further includes a number of I/O pins (not explicitly illustrated). A first subset of which may be employed to couple selected ones of the outputs of reconfigurable interconnects **106** to reconfigurable interconnects of other logic boards (thereby coupling the emulation resources of the logic boards). A second subset of which may be employed to couple data processing resources **102** to certain control resources, such as a control workstation. Accordingly, a novel emulation system may be formed using multiple ones of logic board **100** and control resources, wherein data processing resources **102** of the various logic boards **100** may be employed to locally and correspondingly (i.e. distributively) perform a number of emulation functions on behalf of and at the direction of the control

resources. As a result, the operation efficiency of the novel emulation system is improved.

Emulation ICs **104**, in particular, their on-chip reconfigurable logic and interconnect resources, as in prior art "FPGAs", are used to "realize" the netlists of an IC design to be emulated. In various embodiment, each emulation IC **104** may advantageously include integrated debugging facilities, such as those included with enhanced "FPGAs" described in USP 5,777,489, and co-pending U.S. Patent Application number <insert the CIP number here>, to be described more fully below.

Reconfigurable interconnects **106**, as alluded to earlier, facilitate coupling of the emulation resources of the various emulation ICs **104** of the different logic boards **100** employed to form an emulation system. On-board bus **108** and trace memory **110** perform their conventional functions of facilitating on-board communication/data transfers, and collection of signal states of the various emulation signals of the assigned partition of the IC design being emulated.

In various embodiments, the emulation functions distributively and correspondingly performed on behalf of, and under the direction of the control resources, by on-board data processing resources **102** (of each logic board **100**) include local determination (at the individual board level) of the routing to interconnect the reconfigurable logic resources of the hosted emulation ICs **104** to be used to emulate the circuit elements of the netlists of the partition of an IC design "assigned" to logic board **100** (i.e. distributive routing of an IC design), as well as local generation (at the board level) of configuration signals to configure the emulation resources of the on-board emulation ICs **104** and other on-board emulation resources, such as reconfigurable interconnects **106** (i.e. distributive configuration of emulation resources). Data processing resources **102** distributively

and correspondingly perform these emulation functions responsive to routing and configuration requests from the control resources of the emulation system.

In various embodiments, the emulation functions distributively and correspondingly performed on behalf of, and under the direction of the control resources, by on-board data processing resources **102** (of each logic board **100**) include local generation (at the board level) of testing stimuli, and application of the generated testing stimuli to the appropriate "input pins" of the IC design being emulated. In various embodiments, the emulation functions distributively and correspondingly performed on behalf of, and under the direction of the control resources, by on-board data processing resources **102** (of each logic board **100**) include local determination (at the board level) of the emulation state elements of an IC design being emulated to be monitored, reading of the state data of the emulation state elements to detect occurrence of certain events, and reporting of the occurrence of the events upon their detection. Data processing resources **102** distributively and correspondingly perform these emulation functions responsive to testing and/or monitor requests from the control resources of the emulation system.

In various embodiments, the emulation functions distributively and correspondingly performed on behalf of, and under the direction of the control resources, by on-board data processing resources **102** (of each logic board **100**) include local pre-processing (at the board level) of captured trace data to determine the signal state of one or more signals of interest of the IC design being emulated. Data processing resources **102** distributively and correspondingly perform these emulation functions responsive to trace data requests from the control resources of the emulation system.

The novel manners in which these emulation functions are performed provide at least the advantage of reducing the amount of control signals and data needed to

be transferred in and out of emulation logic board **100** to configure emulation resources of emulation ICs **104** to emulate and debug an assigned partition of an IC design. Additionally, the present invention also provides the advantage of speeding up debugging, testing, including co-simulation of a hardware-software design,
5 performed using an emulation system constituted with multiple ones of logic boards **100**.

On-board bus **108**, reconfigurable interconnects **106**, system bus **108** and trace memory **110**, are intended to represent a broad range of these elements known in the art, accordingly will not be further described. At least one embodiment
10 each for emulation ICs **104** and data processing resources **102** are described in turn below.

Referring now to **Figures 2a-2b**, wherein two block diagrams illustrating an emulation IC **104** in further details, in accordance with one embodiment, are shown.
15 As illustrated in **Fig. 2a**, emulation IC **104** includes reconfigurable LEs (RLR) **202**, reconfigurable interconnects (RIN) **204**, emulation memory (MEM) **206**, debugging resources (DBR) **208**, context or state elements (CTX) **210**, and configuration registers (CR) **212** and **214** coupled to each other as shown. Reconfigurable LEs **202**, emulation memory **206** and context/state elements **210** are used to “realize”
20 circuit elements of the netlists of an assigned partition of an IC design to be emulated. In particular, reconfigurable LEs **202** are used to “realize” the combinatorial logic of the netlists of the assigned partition of an IC design to be emulated. Context/state elements **210** are used to “realize” state elements of the netlist of the assigned partition of the IC design to be emulated, such as flip-flops,
25 and so forth, whereas emulation memory **206** are used to “realize” storage elements of the netlists of the assigned partition of the IC design to be emulated.

Reconfigurable interconnects **204** are used to reconfigurably couple LEs **202**, memory **206** and so forth.

In various embodiments, configuration of these elements, including determination of the interconnect routing, to emulate the netlists of an assigned partition of an IC design, reading of state data of state elements, including determination of which state elements to read, capturing of signal states, including re-creation of "unobservable" signals, are locally (i.e. distributively) performed by data processing resources **102** of the host logic board **100**, to be described more fully below.

Figure 2b illustrates certain aspects of debugging resources **208** of emulation IC **104** in further details in accordance with one embodiment. As illustrated, debugging resources **208** of an emulation IC **104** include scan memory **224** and reconfigurable interconnect **222** reconfigurably coupling scan memory **224** to the reconfigurable logic resources **202** of emulation IC **104**. Scan memory **224** outputs to on-board trace memory **110**.

Scan memory **224** is designed to operate responsive to a debug clock that is typically faster than the emulation clock. Accordingly, during one emulation clock cycle, scan memory **224** may enable signal states of different signals of the netlists of an assigned partition of an IC design being emulated to be captured.

Except for the novel manner on-board data processing resources **102** control and operate these enumerated elements of emulation IC **104**, the various enumerated elements of emulation IC **104** are otherwise known in the art, accordingly will not be further described.

Referring now to **Figure 3**, wherein a block diagram illustrating on-board data

processing resources **102** of one logic board **100** in further details, in accordance with one embodiment, is shown. As illustrated, data processing resources **102**, for the embodiment, include processor **302**, dynamic random memory **304**, and I/O interface **306**, coupled to each other as shown. Processor **302**, memory **304** and I/O interface **306** are also coupled to the earlier described logic board bus **108**.

Memory **304** is used to store a “working” copy of the various software for performing the local (i.e. distributive) routing determination, configuration signals generation, event detection, test stimuli generation, as well as pre-processing of captured signal states, and so forth. For the illustrated embodiment, the software is downloaded to memory **304** during initialization. For this embodiment, the download is preferably staged with the software associated with interconnect routing determination and configuration signal generation being downloaded first, while the software associated with distributed debugging and testing operations download later. In alternate embodiment, logic board **100** may additionally be provided with non-volatile storage, such that a “permanent” copy of a subset or all of the requisite software may be stored. Processor **302** is used to execute the software and effectuate performance of the aforementioned functions, whereas I/O interface **306** facilitates communication between processor **302** and other entities external to logic board **100**. Except for the novel use of these elements to effectuate provision of the desired advantages, these elements of data processing resources **102** are otherwise known in the art, accordingly will not be further described.

Figure 4 illustrates an overview of the software modules provided to data processing resources **102** to provide the desire functionalities of the present invention to a logic board, in accordance with one embodiment. As illustrated, software **400** includes control module **402**, and functional modules, router **403**,

configurator **404**, trace data processor **406**, state data/event detector **408**, and test stimuli generator **410**. Control module **602** is equipped with control logic to facilitate the overall operation in the delivery of the desired functions, including in particular communication with external entities outside a logic board **100**, and invocation of appropriate ones of the functional modules. In one embodiment, control module **402** communicates with the external entities on a request and response transaction basis, via communication packets. That is, under the present invention, the assigned netlists, as well as the routing, configuration, signal state, state data of state elements, and testing requests are made, acknowledged and responded to using transaction messages sent and received in a series of communication packets. Of course, in alternate embodiments, other communication techniques may be used instead.

Fig. 5 illustrates an example communication packet, in accordance with one embodiment, is shown. As illustrated, communication packet **500** includes packet header **502**, command field **504**, parameters associated with the specified commands, or pointers to these parameters **506**, and end of packet marking **508**. Packet header **502** and EOP **508** facilitate provision of various communication related control information, as common in the art of communication. Command **504** facilitates communication between processor **302** and the external entities on the tasks to be performed, and their results. Parameters or pointers thereto **506** augment the commands or return of results, where appropriate.

Referring back to **Fig. 4**, router **403** is equipped with logic to locally (therefore, correspondingly and distributively) determine the routing for

interconnecting the reconfigurable resources of emulation ICs **104** assigned to emulate circuit elements of the netlists of an assigned partition of an IC design to be emulated, responsive to routing requests. Except for the fact that routing of the reconfigurable resources of emulation ICs **104** of a logic board is locally performed on the “assigned” logic board, the tasks of routing, in and of themselves, are otherwise substantially the same when they are centrally performed on a control workstation of an emulation system, and known in the art.

Configurator **404** is equipped with logic to locally (therefore, correspondingly and distributively) generate the configuration bits necessary to configure the reconfigurable logic and interconnect resources of emulation ICs **104**, and the applicable interconnect resources of the logic board, in accordance with the reconfigurable logic resources and board level interconnect centrally determined and the reconfigurable interconnect within the emulation ICs **104** locally determined, responsive to configuration requests. Similarly, except for the fact that generation of the configuration bits is locally performed on the “assigned” logic board, the tasks of generating configuration bits in accordance with a resource allocation, in and of itself, is otherwise substantially the same as the tasks are centrally performed on a control workstation of an emulation system, and also known in the art.

Trace data processor **406** is equipped with logic to locally pre-processes the captured signal states of the emulation signals to determine one or more signal states of one or more signals of interest of the netlists of the assigned partition of the IC design being emulated, responsive to trace data requests. Again, except for the fact that the captured signal states of the emulation signals are pre-processed locally, reconfiguration of debugging resources and processing of capture signal states, in and of themselves, are otherwise substantially the same as these tasks

are centrally performed on a control workstation of an emulation system, and also known in the art.

State data/event monitor **408** is equipped with logic to monitor emulation state elements to detect one or more events. State data/event monitor **408** is further equipped to report the occurrences of the events upon detecting their occurrences. Test vector generator/applicator **410** is equipped with logic to locally (therefore, correspondingly and distributively) generate and apply testing stimuli to the netlists of the assigned partition of the IC design being emulated. Test vector generator/applicator **410** is intended to represent a broad range of testing software known in the art. Similarly, except for the fact that retrieval of state data, event detection, generation and application of test stimuli are locally performed, each of these operations, in and of itself, is substantially the same as the operation is centrally performed at the control workstation.

Referring now to **Figure 6**, wherein a block diagram of an emulation system formed using logic boards **100** incorporated with the teachings of the present invention, in accordance with one embodiment, is shown. As illustrated, emulation system **600** includes control workstation **602** and emulator **606**. Control workstation **602** is equipped with EDA software **604**. Emulator **606** includes a number of logic boards **100**, each having a number of emulation ICs **104** and on-board data processing resources **102** disposed there on as described earlier. In addition to logic boards **100**, emulator **606** also includes service and I/O boards **608**. Boards **100** and **608** are interconnected by inter-board interconnects **610**. In one embodiment, various boards **100** and **608** are packaged together to form a crate (not shown), and the crates are interconnected together via inter-board interconnect **610**. The precise numbers of emulation ICs **104** disposed on each board, as well as

the precise manner in which the various boards are packaged into crates are unimportant, and application dependent.

EDA software **604** is incorporated with the teachings of the present invention, in particular, the corresponding distribution of emulation logic board level routing of the interconnects for netlists of the various partitions of an IC design to be emulated, and configuration of the emulation resources of the emulation ICs of a logic board, to the logic boards themselves. Except for the teachings of the present invention incorporated, EDA software **604** is otherwise intended to represent a broad range of the software typically supplied with an emulation system, including in particular the software for partitioning the netlists of an IC design to be emulated at the system level, and the software for debugging and testing an IC design being emulated, such as model simulators.

Similarly, except for the employment of emulation logic boards **100**, thereby providing the desired advantages, emulator **606** is also intended to otherwise represent a broad range of emulators known in the art.

Referring now to **Figures 7-9**, wherein three flow diagrams illustrating the essential flows of the methods of the present invention for locally and correspondingly performing a number of emulation functions by distributed data processing resources **102** of various logic boards **100** on behalf of and under the direction of the control resources of an emulation system, in accordance with three embodiments, are illustrated. For these embodiments, in addition to the provision of the necessary software functions, such as those illustrated referencing **Fig. 4**, distributed data processing resources **102** of various logic boards **100** are also provided with, either at initialization, in conjunction with the requests or combinations thereof, the netlists of their assigned partitions of the IC design to be emulated,

including their state as well as combinatorial circuit elements, the mappings between the emulation circuit elements and the "original" circuit elements, and the mappings between the observable emulation signals and the "unobservable" signals, if transformations were performed on some of the circuit elements. Distributed data processing resources **102** of various logic boards **100** are also provided with the identifications of the reconfigurable logic resources of their emulation ICs **104** to be employed to emulate the various circuit elements of the netlists of the assigned partitions.

Referring now to **Fig. 7**, wherein the method of the present invention for locally and correspondingly determining interconnect routing (within the emulation ICs of a logic board) and generating configuration signals (for board level and within emulation IC interconnects) by distributed data processing resources **102** of various logic boards **100**, in accordance with one embodiment, is shown. As illustrated, the method starts with the EDA software **604** executing on workstation **602** reading an IC design to be emulated, block **702**. EDA software **604** first partitions the netlists of the IC design into partitions to be emulated by the emulation ICs **104** of the various logic boards **100**, assigning the netlists of the various partitions to the various logic boards **100**, block **704**. For the embodiment, EDA software **604** also determines routing on the various logic boards **100** to interconnect the assigned emulation ICs **104** of the logic boards, also block **704**. Further, EDA software **604** provides the various logic boards **100** with the relevant assignment and on-board routing information, block **704**, as well as requests the interconnect routing within the assigned emulation ICs **104** to be determined locally by the corresponding logic boards **100**. Moreover, the configuration bits for configuring the various reconfigurable logic and interconnect resources are to be generated locally by the

corresponding logic boards **100**. If necessary, the provision and request operations are iteratively re-performed.

Upon receipt of the provided assignment and on-board information, and the on-chip routing and configuration requests, the router/configuration software,
5 executed by distributed data processing resources **102** of each of logic boards **100**, locally and correspondingly determine the routing within emulation ICs **104** of logic boards **100** to interconnect the reconfigurable logic resources within the assigned emulation ICs **104** of the logic boards **100** allocated to emulate the circuit elements of the netlists of the partitions of the IC design, block **706**. Again, if necessary, the
10 operation is iteratively re-performed.

Further, distributed data processing resources **102** of each logic board **100** would locally and correspondingly generate the appropriate configuration bits to configure the allocated emulation resources of the hosted emulation ICs **104**, and the on-board resources of logic boards **100** according to the centrally determined
15 partition and board level routing, and locally determined within emulation IC routing, block **708**. Similarly, if necessary, the operation is iteratively re-performed.

As a result, as alluded to earlier, the amount of signals having to be transferred to the various emulation logic boards **100** to configure the emulation resources of their hosted emulation ICs **104** to realize an IC design is
20 advantageously reduced.

Referring now to **Fig. 8**, wherein the method of the present invention for locally and correspondingly preprocess captured trace data of interest by distributed data processing resources **102** of various logic boards **100**, in accordance with one
25 embodiment, is shown. As illustrated, EDA software **604** facilitates a user in submitting various trace analysis requests, block **814**. In response to the

submission of a user request, EDA software **604** determines if additional data from the distributed logic boards **100** are necessary to satisfy the user' request, block **816**. If not, response is provided to the user, block **820**. On the other hand if additional trace data is required, one or more requests for the trace data are
5 provided to the applicable one or ones of distributed data processing resources **102** of logic boards **100**. In one embodiment, the trace data requested may include state values of observable as well as "unobservable" signals. For "unobservable" signals, the requests include identification of the unobservable signals of interest, the relevant observable emulation signals from which the "unobservable" signals may be
10 inferred, and the functions for use to infer the signal states of the "unobservable" signals of interest from the relevant observable emulation signals.

At block **804**, distributed data processing resources **102** of the requested logic boards **100**, upon receipt of the trace data requests, locally and correspondingly re-create the "unobservable" emulation signals of interest. At block
15 **806**, upon re-creating the "unobservable" signals, distributed data processing resources **102** of various logic boards **100** correspondingly pack and compress the requested observable as well as the "unobservable" signals of the logic boards **100** for return to EDA software **604**.

Upon receipt of the packed and compressed observable and "unobservable"
20 signals of interest, EDA software **604** decompresses and unpacks to recover the requested signals of interest. Thereafter, the process continues from block **816** as earlier described.

Packing and compression may be practiced using any of these techniques known in the art.

Referring now to **Fig. 9**, wherein the method of the present invention for locally and correspondingly generating and applying testing stimuli as well as monitoring for occurrence of certain events by distributed data processing resources **102** of various logic boards **100**, in accordance with one embodiment, is shown. As illustrated, EDA software **604**, more specifically, various simulation models are executed on control workstation **602**, block **914**. At block **916**, EDA software **604** determines if certain test stimuli are to be applied and events are to be monitored, including whether test stimuli are to be generated.

If certain stimuli are to be applied and occurrences of certain events are to be monitored, EDA software **604** requests selected ones of distributed data processing resources **102** of various logic boards **100** to apply the stimuli of interest, and perform the monitoring, block **918**. In one embodiment, each of the requests includes the emulation state elements from whose state data the occurrence of the events of interest may be discerned. Further, EDA software **604** determines if the stimuli needs to be generated, if so, the stimuli are generated accordingly.

Thus, upon receipt of the request, distributed data processing resources **102** of requested logic boards **100** access the relevant emulation state data, block **906**. Data processing resources **102** of requested logic boards **100** analyze the retrieved state data to detect occurrences of the events of interest, block **908**. Upon detecting one or more occurrences of the events of interest, distributed data processing resources **102** of the detecting logic boards **100** report the detection or detections accordingly, block **910**. Thereafter, the process proceeds as earlier described.

Referring now to **Figure 10**, wherein a block diagram illustrating an emulation IC **104'**, in accordance with an alternate embodiment, is shown. As illustrated, emulation IC **104'** includes reconfigurable LEs (RLR) **202**, reconfigurable

interconnects (RIN) **204**, emulation memory (MEM) **206**, debugging resources (DBR) **208**, context or state elements (CTX) **210**, and configuration registers (CR) **212** and **214** coupled to each other as before (i.e. through on-chip bus). However, for this embodiment, emulation IC **104'** additionally include data processing resources **1002** and trace memory **1004**. Further, debugging resources (DBR) **208** are directly coupled to on-chip trace memory **1004**.

Data processing resources **1002** is provided to supplement data processing **102** provided to logic board **100**. Selected ones of the software components illustrated in **Fig. 4** are provided to data processing resources **1002** where the corresponding functions are executed on emulation IC **102** in lieu of being executed on host logic board **100**. Similarly, trace memory **1004** is provided to augment trace memory **110** provided to host logic board **100**. In some embodiment, trace memory **1004** of various emulation ICs **104'** completely replace trace memory **110** of a host logic board **100**. That is, for these embodiments, trace memory **110** is no longer provided to logic board **100**.

Except for their disposition on emulation IC **102**, their constitutions and functions are substantially the same as their earlier described board disposed counterpart.

Thus, a novel emulation IC, logic board, emulation systems constituted with such logic boards, and various emulation related methodologies have been described. While the methods and apparatuses of the present invention have been described in terms of the above illustrated embodiments, those skilled in the art will recognize that the various aspects of the present invention are not limited to the embodiments described. The present invention can be practiced with modification

and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of restrictive on the present invention.

Reblewski - Emulation Components
And System Including ...